

HAND WRITTEN ODIA
CHARACTER RECOGNITION

A THESIS SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND INSTRUMENTATION ENGINEERING

BY

ANURAAG HOTA (108EI017)

SOURAMYA PRADHAN (108EI022)



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY
ROURKELA

2012

HAND WRITTEN ODIA CHARACTER RECOGNITION

A THESIS SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND INSTRUMENTATION ENGINEERING

BY

ANURAAG HOTA (108EI017)
SOURAMYA PRADHAN (108EI022)

UNDER THE GUIDANCE OF: Prof.(Dr.) SUKADEV MEHER



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY
ROURKELA**

2012



NATIONAL INSTITUTE OF TECHNOLOGY
ROURKELA

CERTIFICATE

This is to certify that the thesis entitled “***Hand Written Odia Character Recognition***” submitted by Anuraag Hota and Souramya Pradhan in partial fulfilment of the requirements for the award of Bachelor of Technology Degree in ***Electronics and Instrumentation Engineering*** at the National Institute of Technology, Rourkela (Deemed University) has been carried out by them under my supervision.

Date:

Rourkela

Prof.(Dr.) Sukadev Meher
Dept. of E.C.E, NIT Rourkela

Rourkela-769008

ACKNOWLEDGEMENT

First of all we would like to thank our Guide Prof.(Dr.) Sukadev Meher under whose esteemed guidance and supervision this work was completed successfully. His patience and caring approach towards us whenever we went astray helped us a lot.

Next we would like to thank all the M.Tech students and the faculty present in the Image Processing Laboratory for their support and timely help.

Above all, we would like to thank all our friends and family members whose direct and indirect support helped us complete our project in time. The thesis would have been impossible without their perpetual moral support.

Date:

Rourkela

Anuraag Hota

Souramya Pradhan

TABLE OF CONTENTS

CHAPTER 1	2
INTRODUCTION	2
1.1 Motivation of the Project	3
1.2 Objective of the Project	4
1.3 Organization of the Project Report	4
 CHAPTER 2	 6
LITERATURE REVIEW.....	6
2.1 Overview	6
2.2 Knowledge Gap	7
 CHAPTER 3	 9
OVERVIEW OF OCR	9
3.1 Pre Processing	10
3.1.1 BINARIZATION.....	11
3.1.2 INVERSION	12
3.1.3 SKELETONIZATION	13
3.2 Feature Extraction	14
3.3 Classification	15

CHAPTER 4	18
IMPLEMENTATION	18
4.1 Pre Processing Steps	18
4.2 Feature Extraction (Zoning).....	19
4.3 Image Recognition (Euclidean Distance Method)	26
4.4 Image Segmentation	27
 CHAPTER 5	 29
SIMULATION AND RESULTS	29
5.1 Simulation Results	29
5.2 Recognition Accuracy of each Character	31
5.3 Feature vector and RMSE analysis	34
5.4 Image Segmentation Results	50
5.5 Comparative Study	52
 CHAPTER 6	 54
CONCLUSIONS AND FUTURE WORKS	54
6.1 Conclusions	54
6.2 Scope for Future Works.....	54
 BIBLIOGRAPHY.....	 55

List of Figures

[FIGURE 1 : STEPS FOR OCR]	9
[FIGURE 2: STEPS OF PREPROCESSING].....	10
[FIGURE 3: VARIOUS TYPES OF IMAGES]	12
[FIGURE 4: INVERSION OF THE BINARIZED IMAGE]	13
[FIGURE 5: BINARY IMAGE → INVERTED IMAGE → SKELETONIZED IMAGE]....	18
[FIGURE 6: ZONING]	19
[FIGURE 7: STARTERS, INTERSECTIONS, PSEUDO STARTERS OF 'DA'].....	21
[FIGURE 8: 3 X 3 MASK].....	21
[FIGURE 9: IMAGE MATRIX]	24
[FIGURE 10: LIST OF ODIA CHARACTERS]	29
[FIGURE 11: SIMILAR LOOKING CHARACTERS].....	49
[FIGURE 13: WORD SEGMENTATION]	50
[FIGURE 14: CHARACTER SEGEMENTATION]	51

List of Tables

[TABLE 1: CHARACTERS SHOWING EXACT RESULT]	29
[TABLE 1: CHARACTERS SHOWING EXACT RESULT (cont.)].....	30
[TABLE 2: CHARACTERS SHOWING LESS ACCURATE RESULT]	30
[TABLE 3: CHARACTERS SHOWING LEAST ACCURATE RESULT]	30
[TABLE 4: ACCURACY PERCENTAGE OF EACH CHARACTER].....	33
[TABLE 5: CHARACTERS AND THEIR POSITION IN THE DATABASE]	36
[TABLE 6: FEATURE VECTORS OF TEST CHARACTER 1].....	36
[TABLE 7: FEATURE VECTORS OF DATABASE ‘KA’ CHARACTER].....	37
[TABLE 8: FEATURE VECTORS OF TEST CHARACTER 2].....	39
[TABLE 9: FEATURE VECTORS OF DATABASE ‘KHA’ CHARACTER]	39
[TABLE 10: FEATURE VECTORS OF TEST CHARACTER 3].....	41
[TABLE 11: FEATURE VECTORS OF DATABASE CHARACTER ‘THA’]	41
[TABLE 12: FEATURE VECTORS OF TEST CHARACTER 4].....	43
[TABLE 13: FEATURE VECTORS OF DATABASE CHARACTER ‘DHA’]	43
[TABLE 14: FEATURE VECTORS OF TEST CHARACTER 4].....	45
[TABLE 15: FEATURE VECTORS OF DATABASE CHARACTER ‘YYA’]	45
[TABLE 16: FEATURE VECTORS OF TEST CHARACTER 6].....	47
[TABLE 17: FEATURE VECTORS OF DATABASE CHARACTER ‘TTA’]	47
[TABLE 18: CONFUSING CHARACTERS IN ODIA LANGUAGE]	49

ABSTRACT

The world is fast moving towards digitalization. In the age of super-fast computational capabilities, everything has to be made digitalized so as to make the computer understand and thereby process the given information. Optical character recognition is a method by which the computer is made to learn, understand and interpret the languages used and written by the human beings. It provides us a whole new way by which computer can interact with human beings, in their own languages. Hence OCR has been a topic of interest for researchers all around the globe in the past decade and research paper involving OCR is increasing day by day. It is seen that efficient algorithms have increased the speed and accuracy of character recognition. A substantial amount of work has been done on foreign languages such as English , Chinese etc. but very few paper are there for Indian languages baring a few for Hindi and Bengali. Hence our research work was directed towards development of a novel algorithm for Odia character recognition.

Odia is one of the eighteen languages recognized by the Indian constituency. It is also one of the oldest languages and is spoken by more than 44 million people in the state of Odisha. Recognition of this particular language is difficult because of a number of similar looking characters and the presence of complex characters.

A novel technique is proposed and implemented for the feature extraction method where by a set of 81 feature vectors are extracted to uniquely identify a particular character. The recognition is based on finding the minimum error by implementing the Euclidean distance method. After the implementation of the above technique, accuracy was found to be about 70 % which is much better than many techniques earlier available.

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

The world as we know today is very fast moving and highly automated. Technology has become synonymous to automation. All this is because we, humans, have a tendency to do our job faster and in the most efficient way. Hence the more we automate, the easier and faster our work becomes. Next trend in today's fast changing world is digitalization. Since this is the age of computers we want every information available be digitized and stored in the computers, since they have faster computing capabilities. But the problem in digitizing real world information into digital domain is we need to teach the computer specifically about our concerned real world data.

The topic of our discussion here is the Optical Character Recognition (OCR) technique. OCR has been in quite a few research works in the last decade and it is getting much more attention day by day. It is important because by this we can make the computer learn and recognize the regional languages pretty well and if we do that, then it opens a whole new world of endless possibility! Most of the works about OCR pertain to recognition of hand written characters which can serve as the input for the computer directly. Basically researches have been done on common languages such as English, French, Hindi and some other foreign languages such as Chinese, Japanese etc.

The first step in OCR is going back to the roots of the languages and studying the individual characters which make up the language. Each character is unique in many ways and if we can extract unique features of the individual character we can train the computer about that particular character. Each character has different sets of features which can be used

while comparing with a test character. Hence by this way we can make the computer recognize a character.

Our study is focused on the Odia character recognition. Odia is one of the oldest and is the official language of the Odisha state in the Indian constituency. The Odia language consists of 50 different characters out of which 12 are vowels and rest are consonants. Character recognition in this language is particularly difficult because there are many similar looking characters and the ‘combined characters’ are very difficult to segregate. So to make recognition easier we have developed a novel algorithm. We determine the feature vectors with the help of ‘Zoning’ and then recognition is done by finding the ‘Euclidian’ distance between the test character and the character in our database.

1.1 Motivation of the Project

A substantial amount of work has been done in the field of OCR but very little research has been done for the Odia language. Since Odia is our mother tongue hence we decided to do the project on Odia character recognition. Moreover whatever the techniques that have been implemented in the past have not yet provided a reliable output. Hence more refine methods need to developed for improved output. With many different types of algorithms present for different languages, it gave us an opportunity to study various feature extraction and image classification techniques. The techniques that were studied can then be applied to the Odia language so as to get outputs that can be compared to the one’s we get from our algorithm.

1.2 Objective of the Project

The objective of the project is to develop a technique that can efficiently recognize hand written characters of Odia language. Our main emphasis is on the feature extraction part where we have to extract unique features of each character. Hence we concentrate our work on the feature extraction part where we have taken features based on vertical, horizontal, right diagonal, left diagonal line segments of the character.

1.3 Organization of the Project Report

Chapter 1 gives a brief idea regarding the project work and its importance in today's fast changing world.

Chapter 2 is all about work previously done regarding our topic of interest. It basically gives us the idea regarding various techniques and algorithms presently available.

Chapter 3 is a detailed review about the theoretical part of Optical character recognition. It tells us about the steps involved in any OCR system.

Chapter 4 is all about the work being done by us. It gives detailed information about the algorithm we have studied and implemented in our project work.

Chapter 5 shows various simulation and test results which were generated by implementing our proposed algorithm.

Chapter 6 is the conclusive report about our technique and it also gives an insight regarding scope for future works

CHAPTER 2

LITERATURE REVIEW

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

The basic idea behind doing a literature survey is to gain knowledge regarding the related work. As was in our case, a lot of research paper were taken into consideration and studied. The basic steps involved in character recognition were pre-processing, character segmentation followed by designing an efficient classifier [1]. The pre-processing steps were further studied in detail. The step involves binarization, skeletonization, inversion, thinning, noise reduction as well as image segmentation [2] [4]. Our study involved getting a brief idea pertaining to the difference between on-line and off-line character recognition technique. Considerations have to be made while taking into account the nature of the handwriting, survival of the handwriting which will be followed by recognition, interpretation and then identification [3]. Various features of Odia character were also studied in detail so as to make classification easier [5].

Next various research based on feature vector extraction from an individual character were studied. Features form the basic step for efficient character recognition algorithm. An algorithm can be made much more efficient if the features can be extracted effectively. Feature extraction and classification form an all important step in character recognition [6]. There are a variety of feature extraction techniques available. Our survey concluded that, a statistical based feature extraction technique has to be implemented for our Odia character [7]. The zoning method for the feature extraction was studied and found to be the best suited for our purpose. The idea was to divide an image matrix into $N \times N$ zones and extract

features from each zone [8]. After features had been extracted the next step was using these features to make an efficient recognition technique. A number of classifiers were studied and their accuracy was taken into considerations. The hidden Markov model (HMM) for recognition of handwritten character can serve as a good classifier because, the HMM states are not determined a priori, but are determined automatically based on a database of handwritten images [9]. Apart from the above a classifier based on neural network was also studied and the accuracy was found to be quite appreciable [10].

At the end image segmentation and its various techniques were studied. Segmenting a given image into individual line followed by words and then characters is a complex task. The task is difficult because the text may overlap or merge and segregating them is a difficult and complex process. But at the same time to make the OCR more efficient this step has to be followed. Hence image segmentation can be done by either calculating white spaces or water reservoir method [11] [2]. After completing the survey, a brief idea regarding related work was framed. New ideas and techniques which can enhance the accuracy were developed and implemented in the future work.

2.2 Knowledge Gap

After the study it was found that a very few research work has been done for the Odia characters. Mostly character recognition is done for English, Hindi, Chinese, etc. languages. Many feature extraction and classification techniques have been developed but none can be directly implemented to the Odia character because of its curvature and complex characters. Moreover the existing algorithms provide nearly 65 % accuracy. Hence there is a need for enhanced algorithms that can be used and implemented for improved accuracy and better results.

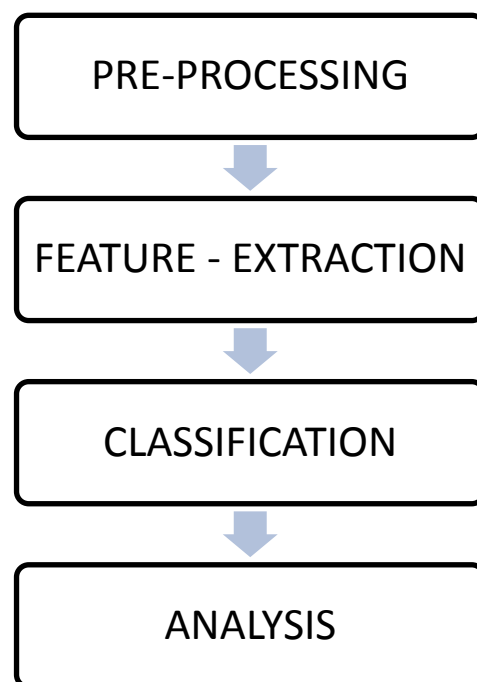
CHAPTER 3

OVERVIEW OF OCR

CHAPTER 3

OVERVIEW OF OCR

Optical character recognition (OCR) is the conversion of handwritten or typed text into an electronic format, which can be stored, interpreted and processed by a computer. It can be used as a direct data input method for a modern computer. Any OCR system is based upon the following four key steps:-

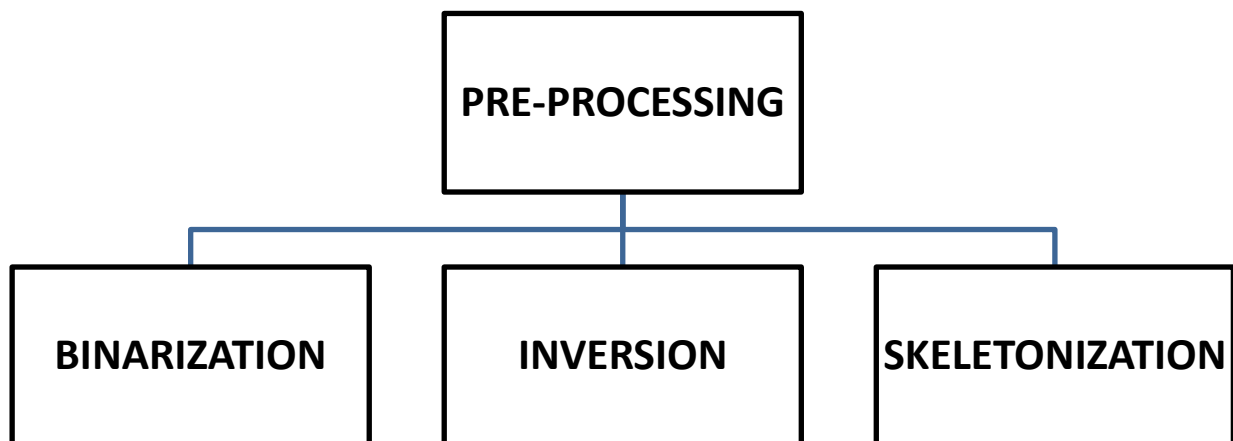


[FIGURE 1 : STEPS FOR OCR]

The Odia character recognition system which is developed also implements the above four steps.

3.1 Pre Processing

Pre-processing is one of the preliminary steps in character recognition. Before the raw data is used for feature extraction it has to undergo certain preliminary processes so that we get accurate results. This step helps in correcting the deficiencies in the data which may have occurred due to the limitations of the sensor. The input for a system may be taken in different environmental conditions. Same object may give different images when taken in different time and conditions. Hence by doing pre-processing we get data that will be easy for the system to operate on, there by producing accurate result.



[FIGURE 2: STEPS OF PREPROCESSING]

3.1.1 BINARIZATION

Matlab image processing toolbox supports four types of images. They are:-

- Gray scale images
- Binary images
- Indexed Images
- RGB images

Gray scale image can be of uint8 or uint16 having values in the range of [0,255] and [0, 65535] respectively. Binary image is a matrix of 0s and 1s. The input image taken is usually a RGB image so the image is converted to a binary image i.e. the image is converted to a black and white image with a particular thresh holding value. The brightness which is the average of the values of red ,green and blue is found for each pixel and is then compared to a threshold value .The pixels which have values less than the threshold value are set to 0 and the pixels whose values are more than the threshold values are set to 1.

The threshold value may be pre-determined or can be calculated from the image histogram. The thresholding can be global or local. In case of global thresh holding, a threshold value is decided based on the entire image. It is often estimated from the histogram level of the image matrix. In case Local thresholding different values are used for each pixel according to local area information

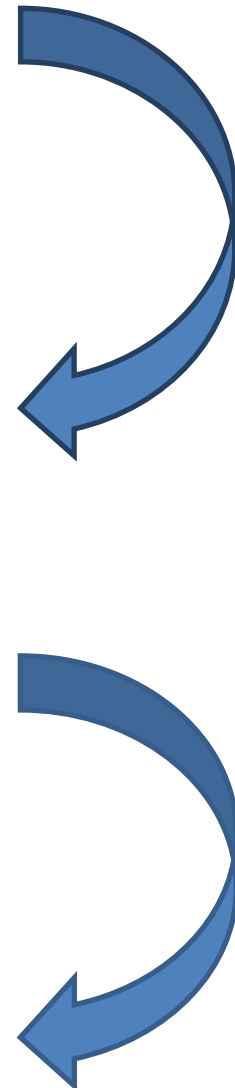
RGB IMAGE



GRAYSCALE IMAGE



BINARY IMAGE

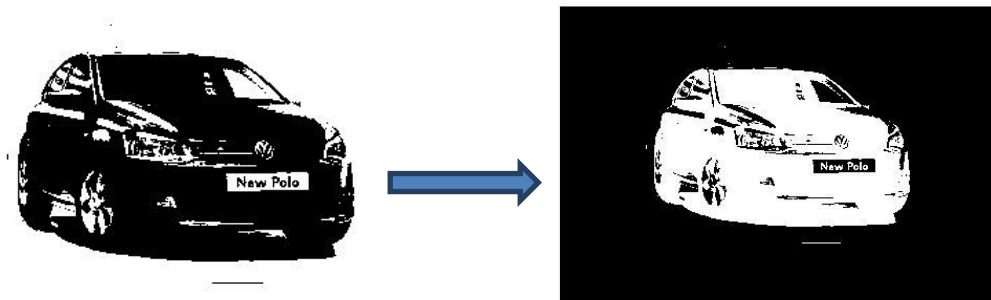


[FIGURE 3: VARIOUS TYPES OF IMAGES]

3.1.2 INVERSION

Inversion is a process by which white pixels are converted to dark and vice versa. The image obtained after binarization contains black pixels in white background. By convention white pixels have value binary 1 and the dark pixels have the value binary 0. So the number of pixels which are 1 far exceeds the number of pixels with 0 values. Since all the processes are

applied on pixel 1 so we need to invert the image so that the calculation reduces significantly there by making the process fast and efficient.



[FIGURE 4: INVERSION OF THE BINARIZED IMAGE]

3.1.3 SKELETONIZATION

The image obtained after inversion is skeletonised, where the foreground pixels are removed preserving the extent and connectivity of the original region. It is useful because it provides a simple and compact representation of the shape of the image. The different methods of skeletonization are as follows:-

- Distance transforms method
- Thinning algorithm

The hit and miss transform is used for the thinning of the image .The thinning with the help of a structuring element .The origin of the structuring element is translated to each and every position of the image. At every position the underlying pixels of the image are compared to the structuring element .If the background and foreground pixels of the image exactly matches that of the structuring element then those pixels are set to 0 otherwise they are left unchanged.

3.2 Feature Extraction

Of all the activities, the feature extraction process is the most important and decisive one because the features extracted determine the efficiency of the classification that follows.

In this stage a set of features are extracted from the character .Maximum features extracted for the character increases the probability of its recognition. Each character is represented as feature vector and those features become the identity for the character. Thus the end result of this process is a feature vector which represents the data of a character for a particular zone.

Feature extraction method is based on the following type's features: -

- Structural
- Statistical.

Structural feature extraction:

It is one of the primitive methods of feature extraction. In this method the features extracted for a character are based on the way the letter is written .It takes into account the curvatures ,edges ,regions ,etc. thus it extracts morphological features. This method sometimes contain features which are not needed for pattern recognition hence this is not an efficient method.

Statistical feature extraction:

In this method the features are extracted from each character is combined to form a feature vector for a particular image. The feature associated with each feature vector is due to the position of the features in the image matrix. The features like length of the line segments, area of the image matrix and the number of particular line segments are obtained for a particular character.

The above two methods use different techniques to extract features from a character. For higher accuracy hybrid techniques, which employ the combination of structural and statistical methods give better results.

The different methods of feature extraction include

- Chain code
- Curve-fitting
- Piecewise –linear regression
- Zoning
- Curve fitting, etc.

3.3 Classification

This step uses a classifier to map a test feature vector to a group consisting of a group of feature vectors among which the discrimination is to be performed .There are many algorithms and methods to that can be used for classification for character recognition. The classifier to be used is decided based on various factors taking into consideration the real world problems. Sometimes also combination of algorithms is used for recognition and it is more effective than using a single classifier.

Classification of the image:

There are many different ways by which the image can be classified. Following are some of the classifiers which are usually used for recognition of features. Apart from this there are many other classifiers available which can be used for specific classification tasks. Each has their own merits and demerits. Their accuracy also varies from each other by a large factor.

➤ **HIDDEN MARKOV MODEL**

It is a statistical model where we are training the system with some unobserved (hidden) states .It consist of a finite set of states where each state is associated with a probability distribution .In a particular state an result or observation can be generated according to the associated probability distribution .The state transition matrix is time independent because the matrix does not change with time and it is and remains constant always .Here we are only able to see the outcome where the states are not visible to an external observer and hence it is called Hidden markov model .

➤ **SUPPORT VECTOR MACHINE**

Here the data is separated into two classes training and testing states .Each object in the training state contains one target value and several other features associated with it. The SVM predicts the target object by comparison with the data attributes. It is one of the best classifier and gives accuracy which is much better than the other classifiers.

➤ **ARTIFICIAL NEURAL NETWORK**

It is a mathematical or computational model which simulates the structure and functional aspects of biological neural network. It is an adaptive system whose structure depends upon external or internal information that flows through the network during the learning phase .The ANN system consists of one input layer, more than one output layer and some intermediate layers. Except the last output layer all other intermediate layers are hidden. The output of the input layer is fed to the hidden layers and then the output from the hidden layers are fed to the output layer .The hidden layers can be trained to give the output image accordingly.

CHAPTER 4

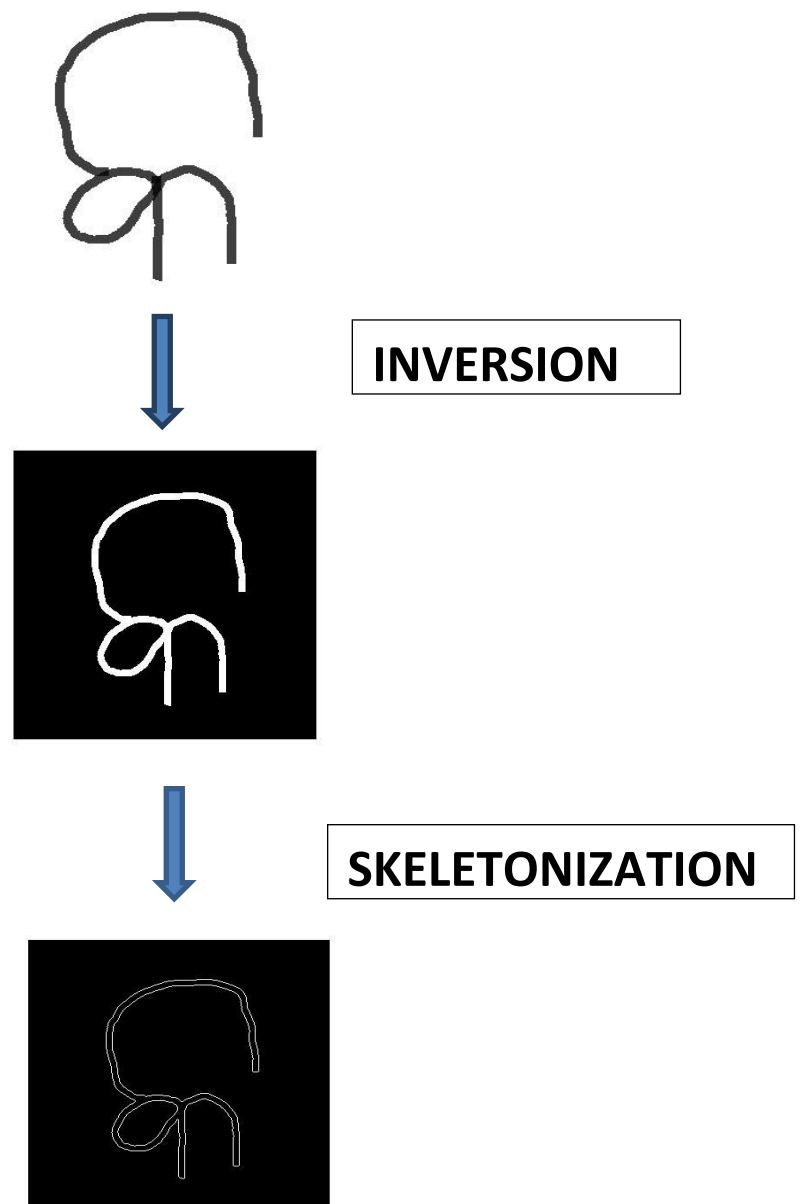
IMPLEMENTATION

CHAPTER 4

IMPLEMENTATION

4.1 Pre Processing Steps

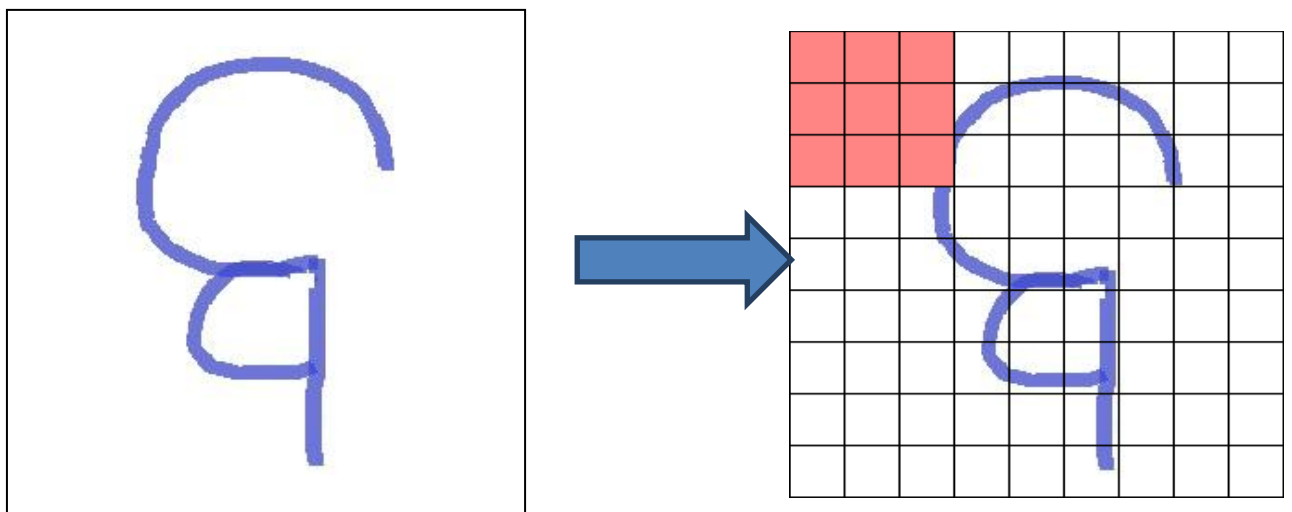
Firstly the test character was input to the OCR system. The image was then pre-processed by following different pre-processing algorithms.



[FIGURE 5: BINARY IMAGE → INVERTED IMAGE → SKELETONIZED IMAGE]

4.2 Feature Extraction (Zoning)

The method used for extraction of features from the image matrix is “Zoning” which is a statistical method of feature extraction. The image matrix taken is divided into windows of equal size, let's say $A \times B$ and the feature vectors are extracted from each of the zones. The image matrix is divided into nine distinct zones for the extraction of features from each of the zones. So here the features are extracted from each of the zones rather than whole of the image. The above method helps us to extract the different line segments from each zone and for that all the pixels present in the image matrix has to be traversed. The following procedure helps us to find the line segments. In order to find the line segments certain pixels are designated as pseudo starters, starters and intersections.



[FIGURE 6: ZONING]

Starters:

The pixels which only have one neighbouring pixel in the image matrix are said to be starters. The position of all the pixels i.e. their coordinates are stored in a certain matrix.

Intersections:

Intersection as a layman's definition should have more than one neighbouring pixel. But in real sense intersection is defined with:-

- Direct pixel
- Diagonal pixels.

Direct pixels are those pixels present in the horizontal and vertical neighbourhood position of the pixel under consideration.

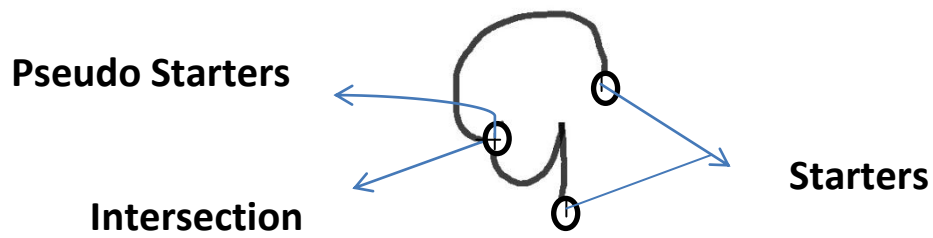
Diagonal pixels are those pixels which are present in the diagonal position of the pixel under consideration.

- If the pixel under consideration has 3 neighbours, for the pixel to be an intersection should have no adjacent diagonal and direct pixels.
- If the pixel under consideration has 4 neighbours then for it to be considered as a intersection it must have none of the direct and diagonal pixels as adjacent.
- If the pixel under consideration has 5 or more neighbours then it is always considered as an intersection.

All the intersections obtained are then stored in a different matrix.

Pseudo Starters:

Pseudo starters are required after all the starters and intersection has been collected .In case of intersections, the pseudo starters are the pixels which are un traversed when moving from a starter to intersection.



[FIGURE 7: STARTERS, INTERSECTIONS, PSEUDO STARTERS OF 'DA']

TRAVERSAL OF THE IMAGE:

The different line segments obtained from the image can be categorised into

- Horizontal line
- Vertical line
- Left diagonal line
- Right diagonal line

These line segments can be determined with the help of a 3 X 3 mask.

4 _(-1,-1)	5 _(-1,0)	6 _(-1,1)
3 _(0,-1)	A _(0,0)	7 _(0,1)
2 _(1,-1)	1 _(1,0)	8 _(1,1)

[FIGURE 8: 3 X 3 MASK]

The pixels of the obtained line segment is analysed with the help of the above mask. The centre pixel of the mask is positioned on each of the pixel of the line segment obtained. The neighbouring pixels are then defined with respect to the central pixel of the mask.

In this mask A is the central pixel. The rest direct and diagonal pixels are numbered in a clockwise manner starting from 1 and the value 1 is given to the pixel below the central pixel. The mask is traversed in the order the pixels are present in the line segment.

The line segment is decided according to the following criterion:

- The line segment is a right diagonal if the maximum values for the pixels of the line segment are 2 or 6.
- The line segment is a left diagonal if the maximum values of the pixels of the line segment are 4 or 8.
- The line segment is a vertical line if the maximum occurring values of the line segment is 1 or 5.
- The line segment is a vertical line if the maximum occurring values of the line segment is 3 or 7.

If two line segments have the values which has occurred the same number of times then the line segment detected first is considered to be the desired line type.

After the type of line segment is known, then the features are found out for each zone. Following are the features obtained for each zone.

- ❖ The number of horizontal line
- ❖ The number of right diagonals
- ❖ The number vertical lines
- ❖ The number of left diagonal lines
- ❖ Normalized length of horizontal line
- ❖ Normalized length of right diagonals
- ❖ Normalized length of vertical lines
- ❖ Normalized length of left diagonal lines
- ❖ Area of the image matrix

The length of the line segment is found out using the following formula:

$$\textbf{LENGTH} = \frac{\textbf{total pixels in the line segment}}{\textbf{total pixels in the zone}}$$

The value which is used to normalizes the line segments can be found by,

$$\textbf{VALUE} = 1 - \left(\left(\frac{\textbf{number of lines}}{\textbf{10}} \right) \times 2 \right)$$

Since the image was divided into 9 X 9 zones and nine features were obtained for each zone, so a total of 81 features were extracted for the entire image matrix for a particular character.

EXAMPLE:-

1	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	1	0
0	0	1	0	0	0	1	0	0
0	0	0	1	0	1	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	1	0	0
0	1	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	1

[FIGURE 9: IMAGE MATRIX]

Let's say we have the above image matrix. The above image after it is divided into 9 zones of size 3 X 3 which is then to be analysed with the help of the mask matrix.

In the above image matrix:-

- **Starters** – [(1,1) , (1,9) , (9,1) , (9,9)]
- **Intersection** – (5, 5)
- **Pseudo Starters** – [(1, 4), (6, 6), (4, 6)]

The line segments obtained are –

- [(1,1), (2,2), (3,3), (4,4)]
- [(9,1), (8,2), (7,3), (6,4)]
- [(4,6), (3,7), (2,8), (1,9)]
- [(6,4), (7,7), (8,8), (9,9)]

Now the mask is taken and traversed through each pixel of the line segments. Suppose for line segment (1, 1), (2, 2), (3, 3), (4, 4) the central pixel of the mask is put on the 1st pixel i.e. (1, 1) of this line segment. With respect to this the next pixel of the line segment i.e. (2, 2) will lie on 8 of the mask. Similarly it is continued for the other pixels and we will get 8, 8 for this line segment and hence it is Left Diagonal.

Similarly, following the above procedure for other line segments we can find which type of line segment it is like

[(9,1), (8,2), (7,3), (6,4)] is a Right Diagonal

[(4,6), (3,7), (2,8), (1,9)] is a Right Diagonal

[(6,4), (7,7), (8,8), (9,9)] is a Left Diagonal .

4.3 Image Recognition (Euclidean Distance Method)

Euclidean distance is the ordinary distance that is generally measured between two points or in our case between two feature vectors. The features extracted are classified based on the method of Euclidean Distance. According to this method, we find the mean square error of the the input image feature vector with 81 features with the feature vector of the images in the database. The feature vector, for which the mean square error is the minimum, is the character in the image.

The RMSE is generally used to measure differences between the values those are predicted by a model and the values actually observed. It serves as a good measure for accuracy. In here, the difference between the feature vectors of the test character and that present in the database is calculated, the minimum of which is found to be the result.

The main advantages of using this technique are:-

- Simple computation
- Can be easily embedded in powerful image recognition techniques
- Insensitive to small deformation

$$RMSE = \sum_{i=1}^{81} \sqrt{T_i^2 + D_i^2 - 2 * T_i * D_i}$$

Where,

T_i =features of the test character

D_i =features of the database characters

4.4 Image Segmentation

Segmentation is a process of separating a digital image into a number of sets of smaller pixels. The main aim of segmentation is to simplify or change the representation of the image into something which is easier to analyse. It is generally used to identify lines, curves etc. in an image. Hence it forms an all important step in image pre-processing. Basically, the input to any OCR system will either be a handwritten manuscript of a typed document. Hence the OCR should first segregate each character individually before starting the recognition step.

There are many algorithms available for image segmentation. Some of them are:-

- Clustering method
- Split and Merge method
- Compression based method
- Histogram Based Method
- Edge detection, etc.

Image segmentation is again of 2 types:

- **Implicit Segmentation**, where words are recognized directly without segmenting them as letters.
- **Explicit Segmentation**, where word is segmented to the smallest possible length available that can even be a character or a 'matra'.

The technique implemented in here is segmentation of an entire paragraph first line by line, which is followed by words and then individual character from the word. The space between each character, words and line has been taken into consideration for segmentation purpose. A threshold value is taken randomly after several simulations so as to justify what must be the minimum value to separate an image as a word or individual character.

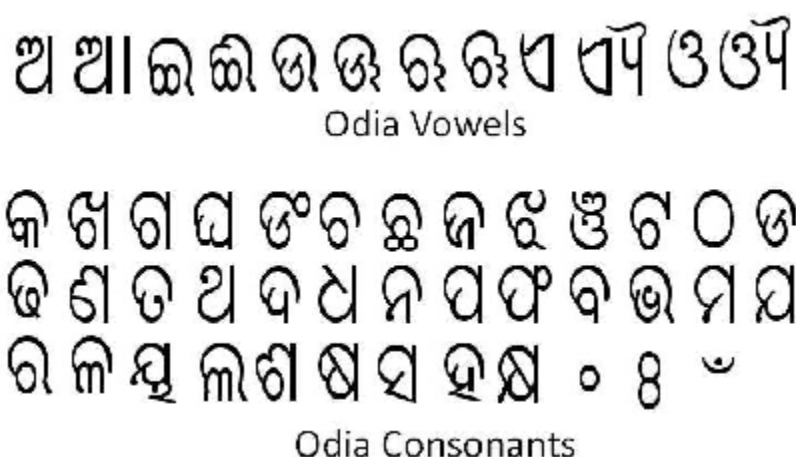
CHAPTER 5

SIMULATIONS AND RESULTS

CHAPTER 5

SIMULATION AND RESULTS

The Odia language consists of 52 character constants which form the entire language. Out of them 12 are vowels and rest are consonants.



[FIGURE 10: LIST OF ODIA CHARACTERS]

Results were obtained by rigorously testing each character 10 times through our proposed algorithm.

5.1 Simulation Results

- Characters showing **exact** result:

ଅ	ଆ	ଇ	ଈ	ଉ
ଓ	ଘ	ଙ	ଚ	ଛ

[TABLE 1: CHARACTERS SHOWING EXACT RESULT]

[TABLE 1: CHARACTERS SHOWING EXACT RESULT (cont.)]

- Characters showing *less accurate* result:

--	--	--	--	--

[TABLE 2: CHARACTERS SHOWING LESS ACCURATE RESULT]

- Characters showing *least accurate* result:

--	--	--	--	--	--

[TABLE 3: CHARACTERS SHOWING LEAST ACCURATE RESULT]

5.2 Recognition Accuracy of each Character

Accuracy was determined by taking 10 test inputs for each individual character. It was computed by the given formula:

$$ACCURACY(IN \%) = \frac{(Total\ no.\ of\ test\ characters - no.\ of\ recognized\ charecter)}{Total\ no.\ of\ test\ characters} \times 100$$

Each character was first run through the algorithm to create the database of all the characters available. After the database was created it was time for testing. Hence test character was run through the algorithm. The RMSE was found between the feature vectors of the test character and each other character individual character present in the database.

The lowest error value signifies the test character matched with the position of the image giving the lowest value. By this method recognition is done in a successful way. In order to test the algorithm, it must be executed with all type of character input.

The more the number of simulation the better is our understanding regarding its efficiency and ability to perform even if an absolutely absurd input is given. After testing on a number of test images a compilation of accuracy for each individual character is listed in the following table.

CHARACTER	RECOGNITION ACCURACY (IN %AGE)	CHARACTER	RECOGNITION ACCURACY (IN %AGE)
ଅ	90	ଝ	80
ଆ	90	ଞ	70
ଇ	90	ଟ	90
ଈ	80	ଠ	80
ଉ	90	ଡ	80
ଋ	80	ଣ	70
ୠ	90	ତ	70
ଏ	90	ଥ	90
ଐ	70	ଦ	80
ଓ	70	ଧ	80

CHARACTER	RECOGNITION ACCURACY(IN %AGE)	CHARACTER	RECOGNITION ACCURACY(IN %AGE)
୩	90	୩	70
୩	90	୩	60
୩	80	୩	50
୦	90	୩	40
୩	40	୩	50
୩	60	୩	60
୩	30	୩	70









[TABLE 4: ACCURACY PERCENTAGE OF EACH CHARACTER]









5.3 Feature vector and RMSE analysis







The feature vectors for each character in the data base were found by using the proposed algorithm and were subsequently stored in the database. Then the feature vector of the test character was obtained and then compared with the characters in the database by using the Euclidean distance method to find the actual test character.

The following table shows the position of characters in the database which is used in subsequent simulation.

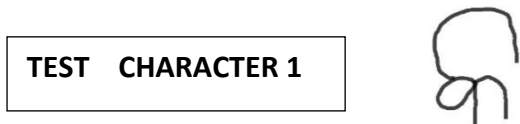
CHARACTER	POSITION	CHARACTER	POSITION
	1		11
	2		12
	3		13
	4		14
	5		15
	6		16

	7		17
	8		18
	9		19
	10		20

CHARACTER	POSITION	CHARACTER	POSITION
	21		28
	22		29
	23		30
	24		31

	25		32
	26		33
	27		34

[TABLE 5: CHARACTERS AND THEIR POSITION IN THE DATABASE]



Feature vector of the TEST character 1:

```
>> a
a =
```

0.8	0.8	0.8	0.8	0.223256	0.474419	0.004651	0.27907	0.01669
0.8	1	0.8	0.8	0.538095	0	0.209524	0.238095	0.016302
1	0.8	0.8	1	0	0.682081	0.306358	0	0.01343
0.8	0.8	1	1	0.464286	0.52381	0	0	0.013041
1	0	0.2	0.2	0	0.465823	0.291139	0.207595	0.030663
1	0.8	1	0.8	0	0.461078	0	0.526946	0.012964
0.8	0.8	0.8	1	0.296296	0.296296	0.388889	0	0.012576
0.6	0.8	0.6	1	0.384083	0.269896	0.32872	0	0.022434
0.8	0.8	1	0.8	0.344828	0.344828	0	0.293103	0.013507

Feature vector of the ‘KA’ in database:

0.8	1	0.8	0.8	0.082418	0	0.565934	0.335165	0.014128
0.6	1	0.8	0.6	0.642857	0	0.190476	0.142857	0.016302
1	0.8	0.6	1	0	0.653846	0.32967	0	0.014128
0.8	0.6	1	1	0.450867	0.531792	0	0	0.01343
0.6	0.8	0.2	0.2	0.07554	0.107914	0.284173	0.359712	0.02158
1	0.8	1	0.8	0	0.529762	0	0.458333	0.013041
0.8	0.6	0.8	1	0.295455	0.346591	0.335227	0	0.013662
0.4	0.6	0.6	0.6	0.351613	0.490323	0.070968	0.054839	0.024065
1	0.8	1	0.8	0	0.342857	0	0.645714	0.013585

[TABLE 7: FEATURE VECTORS OF DATABASE ‘KA’ CHARACTER]

RMSE Values:

>> d

d =

Columns 1 through 4

7.0398 11.5426 10.6479 +13.5610i 14.0819

Columns 5 through 8

12.5372 14.2269 9.8833 12.8958

Columns 9 through 12

15.1208 13.5460 8.9927 12.2430

Columns 13 through 16

13.6038 12.6626 11.6437 13.3370

Columns 17 through 20

12.4265	10.8517	12.7854	13.5404
---------	---------	---------	---------

Columns 21 through 24

9.6348	13.6264	12.3061	11.7844
--------	---------	---------	---------

Columns 25 through 28

14.2342	13.9950	11.0003	10.7169
---------	---------	---------	---------

Columns 29 through 32

12.0170	12.1376	13.7600	11.7582
---------	---------	---------	---------

Columns 33 through 35

10.6331	11.7299	15.2536
---------	---------	---------

>> **minimum**

minimum =

7.0398

>> **position**

position =

1 1

TEST CHARACTER 2

89

Feature vector of the TEST character 2:

>> a

a =

0.8	0.6	0.8	0.6	0.412245	0.191837	0.134694	0.232653	0.019019
0.6	0.8	0.8	0.8	0.353535	0.525253	0.070707	0.020202	0.01537
1	0.8	0.8	1	0	0.592593	0.395062	0	0.012576
0.8	0.4	0.4	0.6	0.216495	0.508591	0.103093	0.137457	0.02259
1	1	0.6	0.6	0	0	0.520202	0.459596	0.01537
1	0.8	1	1	0	0.99115	0	0	0.008772
0.6	0.2	0.8	0.8	0.371517	0.343653	0.114551	0.074303	0.025074
0.4	0.8	1	1	0.638889	0.333333	0	0	0.013973
0.8	0.8	1	0.8	0.331395	0.331395	0	0.313953	0.013352

[TABLE 8: FEATURE VECTORS OF TEST CHARACTER 2]*Feature vector of the 'KHA' in database:*

0.8	1	0.8	0.6	0.637681	0	0.057971	0.285024	0.016069
0.6	0.8	0.6	0.8	0.42126	0.338583	0.188976	0.027559	0.019717
1	0.8	0.8	1	0	0.547771	0.43949	0	0.012188
0.6	0.6	0.8	0.4	0.308725	0.352349	0.137584	0.174497	0.023133
1	0.8	0.6	0.8	0	0.137931	0.374384	0.46798	0.015758
1	0.8	1	1	0	0.99115	0	0	0.008772
0.4	0.4	0.6	0.8	0.282759	0.375862	0.237931	0.072414	0.022512
0.8	0.8	0.8	0.6	0.330275	0.270642	0.068807	0.307339	0.016923
0.8	0.8	1	0.8	0.364641	0.364641	0	0.248619	0.014051

[TABLE 9: FEATURE VECTORS OF DATABASE 'KHA' CHARACTER]

RMSE Values:

>> d

d =

Columns 1 through 4

11.8437 6.3681 8.9126 +13.7697i 14.6168

Columns 5 through 8

11.3896 12.7834 15.1582 13.1866

Columns 9 through 12

14.8296 10.5683 13.3918 11.9832

Columns 13 through 16

9.0047 14.2822 13.8561 16.4972

Columns 17 through 20

16.0023 13.0544 11.5126 10.2973

Columns 21 through 24

13.8067 14.2651 12.0129 10.7855

Columns 25 through 28

11.2525 15.0086 14.4754 12.2785

Columns 29 through 32

13.1643 12.3513 12.2492 12.4585

Columns 33 through 35

8.8060 10.6370 13.1620

>> minimum

minimum =

6.3681

>> position

position =

1 2

TEST CHARACTER 3


Feature vector of the TEST character 3:

>> a

a =

0.8	1	0.6	0.4	0.580645	0	0.080645	0.314516	0.019252
0.8	0.8	1	1	0.174419	0.813953	0	0	0.013352
1	0.8	0.8	1	0	0.503268	0.48366	0	0.011877
1	0.6	0.8	0.8	0	0.661417	0.204724	0.11811	0.019717
0.8	0.6	0.8	0.8	0.085106	0.425532	0.06383	0.382979	0.010946
1	0.8	1	1	0	0.99115	0	0	0.008772
1	0.8	0.8	1	0	0.311377	0.676647	0	0.012964
0.6	0.8	1	1	0.627119	0.355932	0	0	0.01374
1	0.8	0.8	0.8	0	0.258065	0.258065	0.458065	0.012032

[TABLE 10: FEATURE VECTORS OF TEST CHARACTER 3]

Feature vector of the 'THA' in database:

0.8	1	0.6	0.4	0.575107	0	0.06867	0.330472	0.018087
0.8	0.8	1	1	0.348101	0.639241	0	0	0.012265
1	1	0.8	0.8	0	0	0.4125	0.575	0.01242
0.4	0.4	0.8	0.8	0.514599	0.354015	0.058394	0.043796	0.02127
0.8	0.8	0.8	0.8	0.278351	0.293814	0.190722	0.21134	0.01506
1	0.8	1	1	0	0.99115	0	0	0.008772
0.8	0.8	0.6	1	0.304348	0.304348	0.369565	0	0.014283
0.6	0.8	1	0.8	0.560606	0.358586	0	0.060606	0.01537
1	0.8	0.8	0.8	0	0.34104	0.34104	0.300578	0.01343

[TABLE 11: FEATURE VECTORS OF DATABASE CHARACTER 'THA']

RMSE Values:

>> d

d =

Columns 1 through 4

13.1007 10.4344 10.6718 + 8.5157i 12.9546

Columns 5 through 8

12.0827 12.5774 12.0346 11.7591

Columns 9 through 12

12.7363 6.2819 11.3164 9.7956

Columns 13 through 16

11.3553 11.5444 10.5397 12.5297

Columns 17 through 20

12.6361 9.7802 9.8205 10.2704

Columns 21 through 24

11.7662 15.5792 11.1695 12.4999

Columns 25 through 28

10.6502 13.4893 13.7977 13.4394

Columns 29 through 32

10.9261 10.7941 10.9852 14.1722

Columns 33 through 35

8.3991 10.7703 14.9470

>> **minimum**

minimum =

6.2819

>> **position**

position =

1 10

TEST CHARACTER 4


Feature vector of the TEST character 4:

>> a

a =

1	0.8	1	0.8	0	0.76087	0	0.228261	0.014283
1	1	0.8	0.8	0	0	0.503356	0.483221	0.011567
1	0.8	0.8	1	0	0.689655	0.298851	0	0.013507
0.6	0.6	0.8	0.8	0.306667	0.493333	0.04	0.133333	0.017466
0.8	0.8	1	0.8	0.238411	0.596026	0	0.145695	0.011722
1	0.8	1	1	0	0.99115	0	0	0.008772
0.8	0.8	0.6	1	0.318627	0.318627	0.343137	0	0.015836
0.4	0.6	1	0.8	0.618644	0.288136	0	0.067797	0.01832
1	0.8	0.8	0.8	0	0.333333	0.333333	0.315789	0.013274

[TABLE 12: FEATURE VECTORS OF TEST CHARACTER 4]

Feature vector of the 'DHA' in database:

0.6	1	1	0.8	0.78673	0	0	0.199052	0.016379
1	1	0.8	0.8	0	0	0.546053	0.440789	0.011799
1	0.8	0.8	1	0	0.658824	0.329412	0	0.013197
0.6	0.4	0.8	0.6	0.26378	0.429134	0.070866	0.204724	0.019717
0.8	0.8	1	0.8	0.216216	0.574324	0	0.182432	0.011489
0.8	0.6	1	1	0.329412	0.652941	0	0	0.013197
0.8	0.8	0.6	1	0.327869	0.327869	0.322404	0	0.014206
0.6	0.6	1	1	0.533654	0.447115	0	0	0.016147
1	0.8	0.8	0.8	0	0.333333	0.327485	0.321637	0.013274

[TABLE 13: FEATURE VECTORS OF DATABASE CHARACTER 'DHA']

RMSE Values:

```
>> d  
d =
```

Columns 1 through 4

```
11.6206      11.7119      10.3014 + 9.7445i 14.2424
```

Columns 5 through 8

```
11.2920      12.9130      11.8971      10.8027
```

Columns 9 through 12

```
13.4589      9.5515      11.7692      4.8570
```

Columns 13 through 16

```
10.0094      11.1697      10.7444      13.2416
```

Columns 17 through 20

```
13.6291      12.3440      9.2316      10.7802
```

Columns 21 through 24

```
11.6935      14.8624      12.1337      12.0015
```

Columns 25 through 28

```
14.0932      13.8473      13.4408      14.0858
```

Columns 29 through 32

```
10.1243      11.3143      12.3198      14.2933
```

Columns 33 through 35

```
7.2780      8.6693      14.3950
```

```
>> minimum
```

```
minimum =
```

```
4.8570
```

```
>> position
```

```
position =
```

```
1 12
```

TEST CHARACTER 5

Feature vector of the TEST character 5:

>> a

a =

1	1	0.8	0.6	0	0	0.608939	0.374302	0.013895
0.6	1	0.8	1	0.935829	0	0.048128	0	0.014516
1	0.8	0.8	1	0	0.64497	0.343195	0	0.013119
0.8	0.8	0.6	0.8	0.349345	0.388646	0.174672	0.065502	0.017777
0.8	0.8	0.6	0.4	0.178862	0.01626	0.268293	0.50813	0.019096
1	0.8	1	1	0	0.99115	0	0	0.008772
0.8	0.4	0.4	1	0.18797	0.296992	0.488722	0	0.020649
-0.4	0.4	0.4	0.4	0.4225	0.0875	0.3425	0.1025	0.031051
0.6	0.8	1	0.8	0.550781	0.210938	0	0.222656	0.019873

[TABLE 14: FEATURE VECTORS OF TEST CHARACTER 4]

Feature vector of the 'YYA' in database:

1	1	0.8	0.8	0	0	0.538462	0.448718	0.01211
0.4	0.8	1	1	0.955801	0.022099	0	0	0.014051
1	0.8	0.8	1	0	0.583851	0.403727	0	0.012498
0.8	0.8	0.6	1	0.324176	0.494505	0.159341	0	0.014128
0.8	0.4	0.4	0.2	0.121212	0.191919	0.441077	0.205387	0.023055
1	0.8	1	1	0	0.99115	0	0	0.008772
0.8	0.6	0.6	0.8	0.109804	0.54902	0.290196	0.027451	0.019795
0	0.4	0.2	0.6	0.359673	0.343324	0.149864	0.103542	0.028489
0.4	0.6	1	0.8	0.617647	0.183007	0	0.176471	0.023754

[TABLE 15: FEATURE VECTORS OF DATABASE CHARACTER 'YYA']

RMSE Values:

>> d

d =

Columns 1 through 5

9.9726	10.4181	6.2745 +11.9324i	10.3387	11.4114
--------	---------	------------------	---------	---------

Columns 6 through 10

10.3734	10.9728	11.3241	11.7249	13.7923
---------	---------	---------	---------	---------

Columns 11 through 15

12.0151	15.2989	10.7314	11.4254	12.6876
---------	---------	---------	---------	---------

Columns 16 through 20

14.9719	12.5696	12.3914	9.7129	13.5748
---------	---------	---------	--------	---------

Columns 21 through 25

10.6222	9.9073	12.2150	7.9940	11.0123
---------	--------	---------	--------	---------

Columns 26 through 30

10.7261	11.9344	12.7647	11.3418	11.4986
---------	---------	---------	---------	---------

Columns 31 through 35

13.3967	6.1047	13.4746	13.4511	9.6116
---------	--------	---------	---------	--------

>> minimum

minimum =

6.1047

>> position

position =

1 32

TEST CHARACTER 6

Feature vector of the TEST character 6:

>> a

a =

1	1	0.8	0.6	0	0	0.635294	0.347059	0.013197
0.8	0.8	0.8	0.8	0.604762	0.147619	0.090476	0.138095	0.016302
1	0.6	0.8	1	0	0.761905	0.22381	0	0.016302
1	0.8	1	1	0	0.99115	0	0	0.008772
0.6	0.8	0.8	0.8	0.150943	0.136792	0.570755	0.117925	0.016457
0.8	0.8	1	0.6	0.220994	0.39779	0	0.353591	0.014051
0.8	1	0.8	0.6	0.415459	0	0.304348	0.26087	0.016069
0.6	1	0.6	0.8	0.558704	0	0.336032	0.08502	0.019174
1	0.8	0.8	0.8	0	0.291925	0.291925	0.397516	0.012498

[TABLE 16: FEATURE VECTORS OF TEST CHARACTER 6]

Feature vector of the 'TTA' in database:

1	1	0.8	0.8	0	0	0.618182	0.369697	0.012809
0.8	1	0.6	0.8	0.635193	0	0.218884	0.128755	0.018087
1	0.6	0.8	0.8	0	0.36612	0.284153	0.327869	0.014206
0.8	0.8	1	1	0.279221	0.707792	0	0	0.011955
0.6	0.8	1	0.6	0.207729	0.115942	0	0.652174	0.016069
1	0.6	1	1	0	0.985714	0	0	0.010868
1	0.8	0.8	0.8	0	0.283019	0.415094	0.283019	0.012343
0.8	1	0.6	0.6	0.565	0	0.205	0.2	0.015526
1	0.8	0.8	0.8	0	0.269231	0.269231	0.442308	0.01211

[TABLE 17: FEATURE VECTORS OF DATABASE CHARACTER 'TTA']

RMSE Values:

>> d

d =

Columns 1 through 4

10.2330 14.0876 10.5346 +10.5636i 12.5332

Columns 5 through 8

12.5570 16.3458 10.9495 11.4244

Columns 9 through 12

11.6651 13.7367 6.4856 13.6197

Columns 13 through 16

11.2525 8.4399 8.7972 13.3049

Columns 17 through 20

9.9282 12.0746 9.5860 14.4471

Columns 21 through 24

9.4087 15.5019 11.2906 10.8928

Columns 25 through 28

14.5852 17.2818 13.0411 13.7490

Columns 29 through 32

10.8902 10.1097 13.2151 14.4492

Columns 33 through 35

14.0854 14.0663 16.0862

>> minimum

minimum =

6.4856

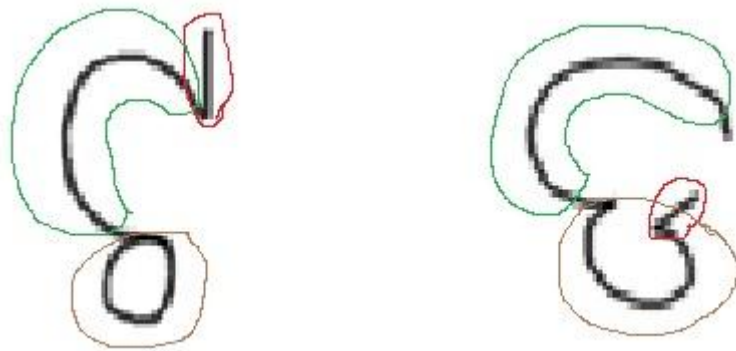
>> position

position =

1 11

The above simulation result for the character ‘TTA’ showed the position of character ‘DA’ which is an error. The RMSE value we obtained here is 6.4856 but the actual result should show the value 8.4399. It was observed that the value 8.4399 is the next RMSE value after 6.4856.

Upon further analysis it was found that the closeness in values is due to the similar shapes of the characters, ‘TTA’ and ‘DA’.



[FIGURE 11: SIMILAR LOOKING CHARACTERS]

Not only this, upon further analysis it was found that there are many other confusing characters in Odia language which had RMSE values very close to each other. The following table list some of the confusing characters in Odia.

ଟ	ଡ
ଟା	ଡା
ଟି	ଡି
ଟୁ	ଡୁ

[TABLE 18: CONFUSING CHARACTERS IN ODIA LANGUAGE]

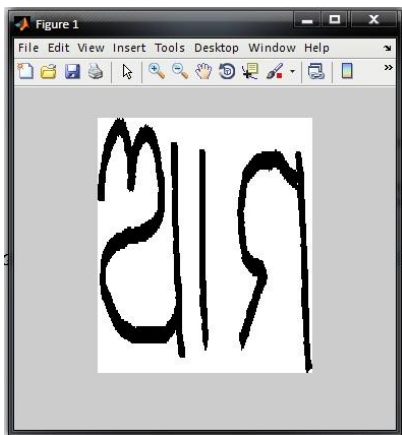
5.4 Image Segmentation Results

Input Image:

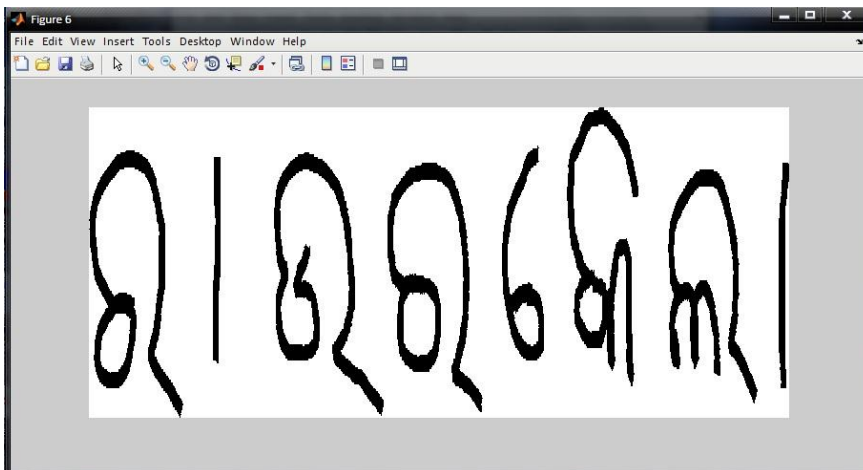
ଆମ
ର । ଚରଚରକର ।

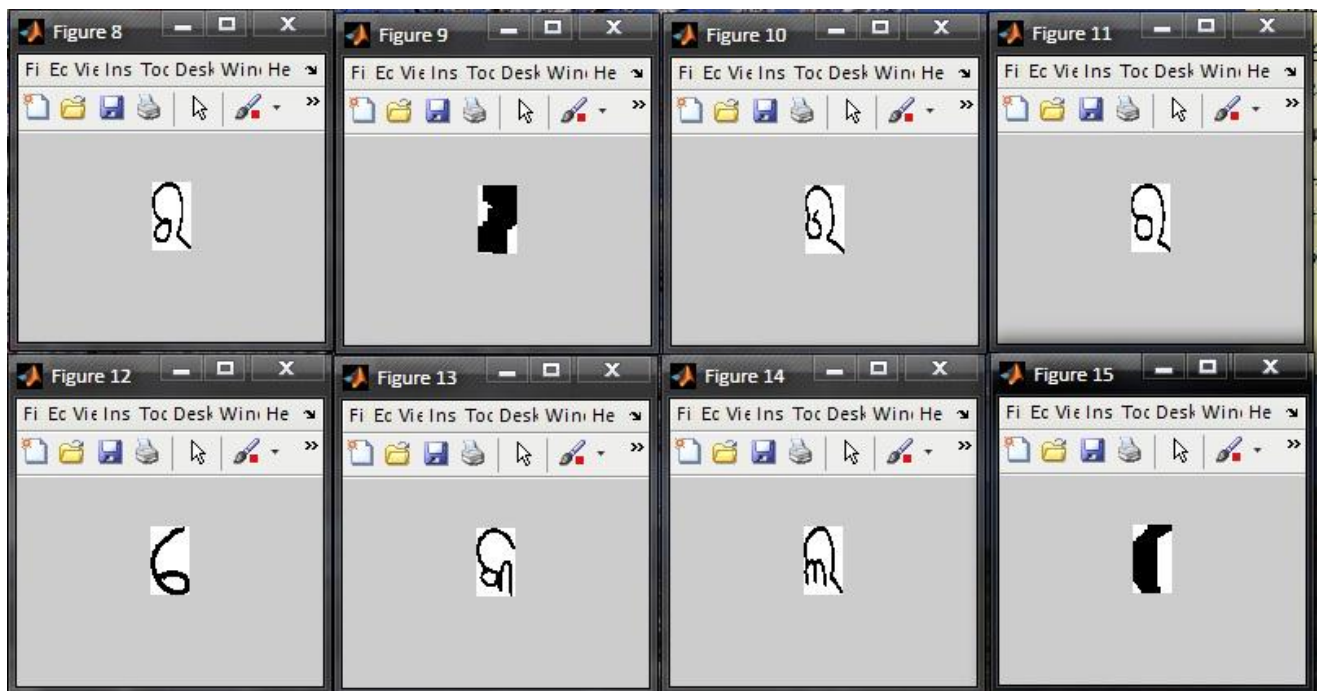
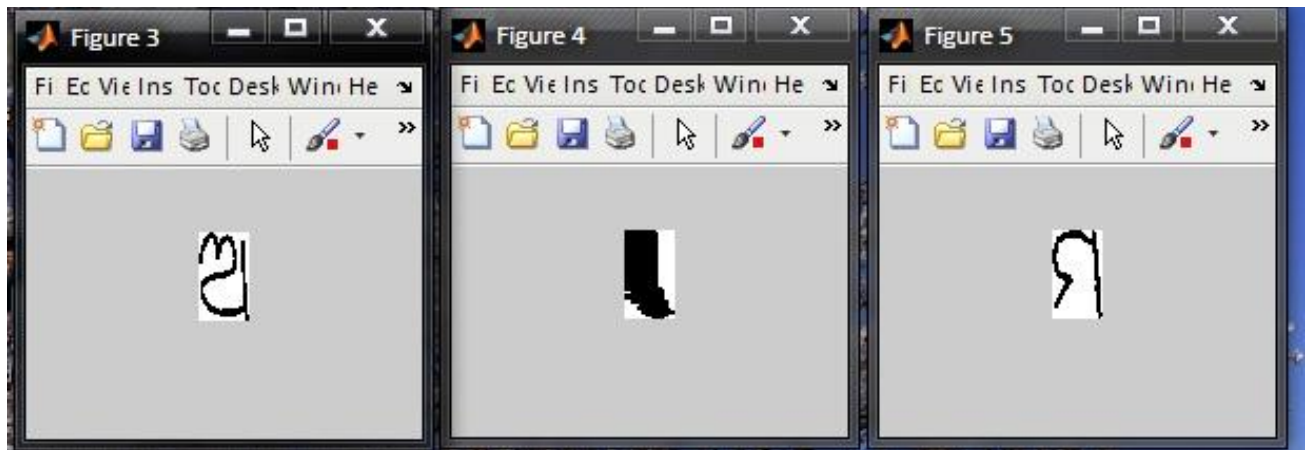
[FIGURE 12: INPUT CHARACTER FOR IMAGE SEGMENTATION]

Output:



[FIGURE 13: WORD SEGMENTATION]





[FIGURE 14: CHARACTER SEGEMENTATION]

5.5 Comparative Study

The accuracy results obtained for other classifiers are as follows:

- Hidden markov Model ----- 56 %
- Naïve Bayes Classifier ----- 63 %
- Neural Network ----- above 90 %
- Support Vector Machine ----- above 90%

But the best among all is the Support vector machine (SVM) which shows better result than all the classifiers even better than Neural Network (NN).

CHAPTER 6

CONCLUSIONS AND FUTURE WORKS

CHAPTER 6

CONCLUSIONS AND FUTURE WORKS

6.1 Conclusions

The hand written Odia character recognition algorithm was successfully tested using a large number of test images. Accuracy was about 70%. Around 20% of the characters deviated from the actual value by a small difference in their RMSE values. Our work was basically focused on envisaging methods that can efficiently extract feature vectors from each individual character. The method we came up with gave efficient and effective result both for feature extraction as well as recognition.

6.2 Scope for Future Works

Once the character has been successfully recognized it opens the door for a lot many possibilities. OCR can be embedded to the computing system so that handwritten or even typed text can serve as direct input for the computer. We can even connect the OCR to a speech processing system which can effectively speak out each character after recognition is done by the OCR. Such a device would be very much useful for persons who are blind. Furthermore Odia Character recognition can be used in the following areas:

- It helps to preserve the old documents in electronic format which can be used for future use
- The large document images can be saved in a limited space.
- It helps the visually impaired persons to to read the content on the document.

BIBLIOGRAPHY

- [1]. G.Vamvakas, B.Gatos, N. Stamatopoulos, and S.J.Perantonis, “A *Complete Optical Character Recognition Methodology for Historical Documents*”, the Eighth IAPR Workshop on Document Analysis Systems
- [2].Debasish Basa ,Sukadev meher , “*Handwritten Odia Character Recognition*”, Presented in the National Conference on Recent Advances in Microwave Tubes, Devices and Communication Systems, Jagannath Gupta Institute of Engineering and Technology, Jaipur, March 4-5,2011
- [3].Rajean Plamondon, Fellow IEEE and Sargur N. Srihari ,Fellow IEEE , “*On-Line and Off-Line Handwritting character Recognition: A Comprehensive Survey*”, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. VOL. 22, NO. 1. JANUARY 2000
- [4]. Sanghamitra Mohanty, Hemanta Kumar Behera , “*A Complete OCR Development System For Oriya Script*”
- [5]. B.B. Chaudhuri U. Pal M. Mitra, “*Automatic Recognition of Printed Oriya Script*”
- [6]. Fumitaka KIMURA, Tetsushi WAKABAYASHI and Yasuji MIYAKE, “*On Feature Extraction for Limited Class Problem*”
- [7]. Supriya Deshmukh, Leena Ragha, “*Analysis of Directional Features - Stroke and Contour for Handwritten Character Recognition*”, 2009 IEEE International Advance Computing Conference (IACC 2009) Patiala, India, 6-7 March 2009
- [8]. M. Blumenstein, B. K. Verma and H. Basli, “*A Novel Feature Extraction Technique for the Recognition of Segmented Handwritten Characters*”, 7th International Conference on Document Analysis and Recognition (ICDAR '03) Eddinburgh, Scotland: pp.137-141, 2003.
- [9]. Tapan K Bhowmik Swapan K Parui Ujjwal Bhattacharya Bikash Shaw, “*An HMM Based Recognition Scheme for Handwritten Oriya Numerals*”.
- [10]. Sung-Bae Cho, “*Neural-Network Classifiers for Recognizing Totally Unconstrained Handwritten Numerals*”, IEEE Transactions on Neural Networks, Vol. 8, No. 1, January 1997
- [11]. N. Tripathy and U. Pal, “*Handwriting Segmentation of Unconstrained Oriya Text*”